Register Identifier:	9

REGISTER OF GRAPHICAL ITEMS
Compression Type Section

Compression Type Name:	PNG Compression Method 0
Sponsoring Authority:	ISO/IEC JTC 1 SC24
Date of Registration:	
ISO Approval Date:	
Amendment Record:	
Description:	

The 'compression type' parameter of the TILE and BITONAL TILE elements indicates the method used to compress the content of the associated tile.

This registered value, 9, indicates that the associated tile is compressed by the technique defined in the ISO/IEC 15948, Portable Network Graphics (PNG) as PNG compression method 0 (zero).

Relationship to standards:

ISO/IEC 8632-1:1992, Information processing systems – Computer graphics – Metafile for the storage and transfer of picture description information Part 1: Functional specification

ISO/IEC 8632-2:1992, Information processing systems – Computer graphics – Metafile for the storage and transfer of picture description information Part 2: Character encoding

ISO/IEC 8632-3:1992, Information processing systems – Computer graphics – Metafile for the storage and transfer of picture description information Part 3: Binary encoding

ISO/IEC 8632-4:1992, Information processing systems – Computer graphics – Metafile for the storage and transfer of picture description information Part 4: Clear text encoding

FCD ISO/IEC 15948:2004, Information technology – Computer graphics and image processing – Portable Network Graphics (PNG): Functional specification

Description

The ‘compression type’ parameter of the TILE and BITONAL TILE elements indicates the method used to compress the content of the associated tile. This value indicates that the associated tile is compressed by the technique defined in the ISO/IEC 15978, Portable Network Graphics (PNG) as PNG compression method 0.

PNG compression method 0 specifies deflate/inflate compression with a 32K sliding window. Deflate compression is an LZ77 derivative. This compression method has been standardized in PNG by W3C (World Wide Web Consortium) as an equivalent and alternative to the LZW compression method, which is commonly used in the GIF raster format. PNG method 0 has no Intellectual Property Rights (IPR) encumbrances, whereas Unisys Corporation holds a patent on LZW and asserts the right to charge license fees for its use.

It is the goal of this registered compression method to enable the transposition of the essential information from a PNG file into a CGM Tile Array element, without requiring the decompression and recompression of the raster content. The transposition of "ancillary information" from the PNG file is also supported.

The compressed content of the TILE element itself, i.e., the Bitstream (BS) parameter, corresponds to the contents of the PNG "IDAT" chunk(s). The following is quoted from clause 10.1 and 10.2 of PNG:

"Deflate-compressed datastreams within PNG are stored in the "zlib" format […].

For PNG compression method 0, the zlib compression method/flags code shall specify method code 8 (deflate compression) and an LZ77 window size of not more than 32768 bytes. The zlib compression method number is not the same as the PNG compression method number defined in the `IHDR` chunk […]. The additional flags shall not specify a preset dictionary. […]

The compressed data within the zlib datastream are stored as a series of blocks, each of which can represent raw (uncompressed) data, LZ77-compressed data encoded with fixed Huffman codes, or LZ77-compressed data encoded with custom Huffman codes. A marker bit in the final block identifies it as the last block, allowing the decoder to recognize the end of the compressed datastream. Further details on the compression algorithm and the encoding are given in the deflate specification [RFC-1951].

The check value stored at the end of the zlib datastream is calculated on the uncompressed data represented by the datastream. The algorithm used to calculate this is not the same as the CRC calculation used for PNG chunk CRC field values. The zlib check value is useful mainly as a cross-check that the deflate and inflate algorithms are implemented correctly. Verifying the chunk CRCs provides adequate confidence that the PNG file has been transmitted undamaged.

[start of 10.2][…]The concatenation of the contents of all the `IDAT` chunks makes up a zlib datastream. This datastream decompresses to filtered image data."

It is the single zlib datastream which is the single compressed content (BS) parameter the TILE element. If the TILE content using PNG method 0 in fact did come from a PNG file, then it is the concatenation of the IDAT chunks.

This registration proposal does not require that the "PNG method 0" compressed content (which is identical to zlib method-code 8) comes from translation of a PNG file. However the information encoding of this registered compression type supports the translation from PNG and the transposition of non-critical PNG chunk types into the "Method Specific Parameters" of the CGM Tile Array. In the case that the LZ77-compressed data did not come from a PNG file, then PNG chunks must be simulated in some of the Required and Optional data, below.

Because PNG method 0 is not a tiled format, the remaining description will assume that a single-tile Tile Array is being generated. (Note: there is nothing in this compression method registration to prohibit generation of multi-tile Tile Arrays using this method). Therefore the parameters of BEGIN TILE ARRAY which define the number of tiles in the cell path direction and line progression direction are well defined: 1 and 1. Similarly,

the image offset parameters of the BEGIN TILE ARRAY will reflect that there is no image offset and no border. The cell path directions and line progression directions would be 0 and 270 for a VDC Extent that defines lower-left origin (and the equivalent image orientation for other VDC Extent orientations).

Note: there is nothing in this compression type registration to prohibit generation of multi-tile Tile Arrays using this registered compression type 9. In a multi-tile LX77-compressed Tile Array, care would have to be taken to get consistent and sensible values into the Required and Optional parameter data in the MSP SDR, below.

The number of cells in the cell path and line progression directions come from the PNG IHDR chunk, 'width' and 'height' values. The IHDR 'bit depth' value defines the Cell Colour Precision of the TILE. The 'colour type' value defines the Colour Selection Mode (indexed or direct), and whether the colour model is RGB or RGB-alpha, as well as addition information per the PNG standard.

The remainder of the information needed to decode a Tile Array using this registered compression method is defined in the method-specific-parameters (MSP) parameter of the TILE element, which encodes PNG chunks. The MSP are encoded using the SDR (structured data record) method of CGM:1992.

Each PNG chunk is encoded into a single SDR member, as defined below. The PNG chunk data encoded into the data section of each SDR member consists of: the 4-byte chunk length; the 4-byte chunk type code; then the chunk data; then the 4-byte CRC. See clause 5.3 of PNG 1.0. The presence of the 4-byte PNG chunk type code in each SDR member makes the members self-identifying as to content, therefore optionality of chunks and arbitrary order of chunks is supported when mapping the PNG content into the Tile Array.

Required members (Critical chunks):

1. The IHDR (header) chunk of the PNG file shall always be encoded into the first member of the MSP SDR.

2. The PLTE (palette) chunk, if present in the PNG file (required or permitted by the colour type in the IHDR chunk), shall be encoded into the second member of the MSP SDR.

Optional members (corresponding to the Ancillary chunks of PNG), in any order:

1. bKGD – background colour.
2. cHRM – primary chromaticies and white point.
3. gAMA – image gamma
4. hIST – image histogram
5. pHYs – physical pixel dimensions (if this chunk is present, it shall be used to define the parameters in the BEGIN TILE ARRAY element, cell size in path direction and cell size in line direction).
6. sBIT – significant bits.
7. tEXt – textual data.
8. tIME – image last modification time
9. tRNS – transparency
10. xTXt – compressed textual data

Note that the pHYs chunk is optional in PNG. If not present (or if the values are 0), then there is no inherent, defined size for the cells in the image, and it could be considered to be a device pixel map for whatever is the output device. This concept is not supported in CGM. The cells of any TILE have physical size, as defined in the mandatory parameters "cell size in path/line direction". Therefore a process which transposes PNG without

a pHYs chunk, or with a pHYs chunk with values "0" (zero), into CGM Tile Array must still supply valid cell size values for the BEGIN TILE ARRAY element.

This encoding of the PNG content into the CGM Tile Array preserves all of the information necessary, for example, to easily reconstruct a valid PNG data stream for input to a conforming PNG decoded/viewer.

The PNG compression method is completely defined in the reference:

[RFC-1951] Deutsch, P., "DEFLATE Compressed Data Format Specification version 1.3", RFC 1951, Aladdin Enterprises, May 1996, <URL:ftp://ds.internic.net/rfc/rfc1951.txt>

CGM Information

1) CGM Functional Specification.

The "PNG Compression method 0" is indicated in a TILE element of a Tile Array by the value "9" in the compression type parameter of the TILE.

The mandatory parameters of the BEGIN TILE ARRAY element and the TILE element are derived from the PNG data as described above.

Each of the Required and Optional members (chunks) described above are encoded into an individual member of the SDR which represents the Method Specific Parameters, one SDR member per PNG chunk.

2) CGM Encodings.

The data of the Method Specific Parameters of each TILE is encoded according to the style of the SDR data type defined for Version 3 metafiles in ISO/IEC 8632:1992 (Annex C of Part 1, and appropriate clauses of the Encodings). Each SDR as a whole is treated as a string in the CGM encoding being used.

Each member of the SDR is encoded identically:

- the data type is BS (Bitstream, data type index "20");
- the data count is 1, i.e., there is one BS parameter per member;
- the chunk data (4-byte chunk length; 4-byte chunk type code, chunk data, 4-byte CRC), as described above.

Parameters of type BS need some information to indicate how long is the BS parameter, or to delimit the end of the BS parameter. In this case, the first 4 bytes of the BS parameter is the 4-byte chunk data length, N. Therefore (according to the chunk-length rules of PNG), the total length of the BS parameter is: N+12 octets.